Dynamic Programming and Reinforcement Learning

Introduction (Week 1)

Rainer Schlosser, Alexander Kastius

Hasso Plattner Institute (EPIC)

April 12, 2021

The World is Full of Decision Problems

How to Approach Decision Problems?

Decisions

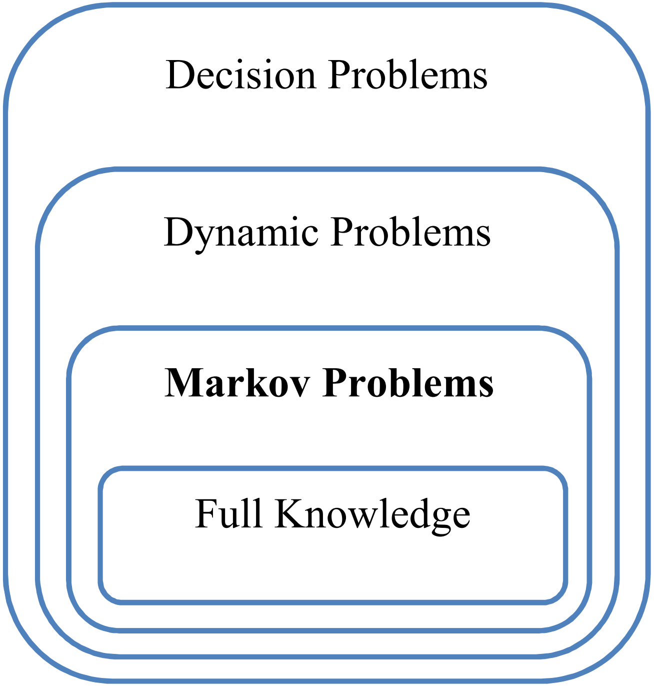
Objectives

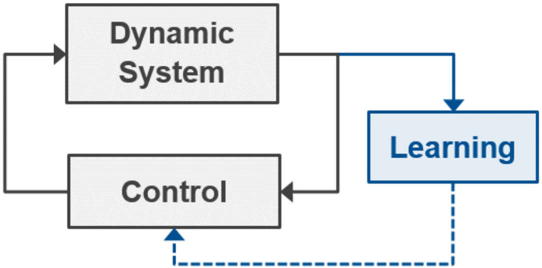
Constraints

Quantify Impact of Decisions

Optimization Approach

Dynamic Problems & Markov Decision Processes (MDP)





Types of Decision Problems

Static vs. **Dynamic**

Deterministic vs. **Stochastic** vs. Uncertainty

Low vs. **Large** Complexity/ Dimensionality

Finite vs. **Infinite** Time

Discrete vs. Continuous Time

Discrete vs. Continuous Actions & Rewards

Risk Neutral vs. Risk-Averse

Markov Type (**yes** / no)

System Dynamics **Known/Unknown**

System Fully vs. Partially Observable

Further Players/Agents

Solution Methods

Optimal	vs.	Heuristic Solutions
• (Exhaustive Search)		• Approximate Dynamic Programming
• Dynamic Programming		• Reinforcement Learning – Q-Learning – Deep Q-Learning – Policy Gradient Algorithms

Agenda

- Introduction ✓
- **Personal Background**
- Structure of the Course & Grading

Personal Background (Rainer)

- Ph.D. Operations Research (2014), Humboldt-University of Berlin
- Hasso Plattner Institute (EPIC) since 2015
- Field of Research
  – Data-driven decision support
  – Focus on stochastic models & Dynamic Programming (DP)
- Current Areas of Applications
  – Revenue management (e.g., dynamic pricing, ordering, advertising)
  – Database configuration (e.g., data placement problems, index selection)

Personal Background (Alex)

- Master Computer Science (2020), HPI
- Hasso Plattner Institute (EPIC), PhD Candidate
- Field of Research
  – Data-driven decision support
  – Focus on Reinforcement Learning (RL)
- Current Areas of Applications
  – Revenue management
  – Database configuration

What about you?

- Background?
- Interests?
- Expectations?
- Questions?

Agenda

- Introduction ✓
- Personal background ✓
- **Structure of the Course & Grading**

Technical Information

- Credits? 4 SWS (V/Ü), 6 ECTS (graded)
- When? Monday 13.30 – 15.00 VL/UE (lecture/exercise)
  Thursday 11.00 – 12.30 VL/UE (lecture/exercise)
  Start: April 12, 2021, End: July 15, 2021
- Where? currently via Zoom (Room: 7271364393, Password: 256757)
- Who? Rainer Schlosser, rainer.schlosser@hpi.de
  Alexander Kastius, alexander.kastius@hpi.de
- Slides? EPIC, Teaching, Summer 2021

Structure of the Course

- April/May: Lectures on Models & Methods:
  - (i) MDPs
  - (ii) Dynamic Programming
  - (iii) Reinforcement Learning
- June/July: Choose Projects, Apply/Extend Suitable Techniques
  Work in Teams, Input/Support will be given
- July/Aug: Documentation of Projects Results

Goals of the Course & Grading

• Goal:	Develop models to compute optimized decisions for different problems & applications	
• Learn:	Optimization techniques	
• Do:	Apply & extend different optimization approaches	
• Grading:	30%	Project results
	70%	Documentation (“Projektarbeit”)
		Deadline Aug 31 (~10-20 pages)

Prerequisites

- Programming

  Parameters, Data Preparation
  Loops, Recursions, Simulations

- Basic Mathematical Background

  Sets, Vectors
  Probabilities, Random Variables, Expected Values

- More does not harm

  Regression Analysis
  Deep Learning
  Experience with Solvers
  Game Theory

Overview

Week	Dates	Topic	
1	**April** 12/**15**	Introduction + **Finite Time Markov Decision Processes (MDP)**	
2	April 19/22	Infinite Time MDPs + Dynamic Programming (DP) Exercise	
3	April 26/29	(Approximate) Dynamic Programming (ADP)	
4	May 3/6	Q-Learning (QL)	
5	May 10	Deep Q-Networks (DQN)	(Thu May 13 "Himmelfahrt")
6	May 17/20	DQN Extensions	
7	May 27	Policy Gradient Algorithms	(Mon May 24 "Pfingstmontag")
8	May 31/June 3	Project Assignments	
9	June 7/10	Work on Projects: Input/Support	
10	June 14/17	Work on Projects: Input/Support	
11	June 21/24	Work on Projects: Input/Support	
12	June 28/July 1	Work on Projects: Input/Support	
13	July 5/8	Work on Projects: Input/Support	
14	July 12/15	Final Presentations	
	Aug 31	Finish Documentation	

What are Dynamic Optimization Problems?

- How to control a dynamic system over time?
- Instead of a single static decision we have a *sequence* of decisions
- The system evolves over time according to a certain dynamic
- The decisions are supposed to be chosen such that
  a certain objective/quantity/criteria is optimized
- Find the right balance between short and long-term effects

Examples Please!

Examples

- Inventory Replenishment
- Selling Airline Tickets
- Drinking at a Party
- Exam Preparation
- Brand Advertising
- Used Cars
- Eating Cake

Task: Describe & Classify

- Goal/Objective
- State of the System
- Actions
- Dynamic of the System
- Revenues/Costs (Rewards)
- Finite/Infinite Horizon
- Stochastic Components

Classification

Example	Objective	State	Action	Rewards	Dynamic
Inventory Mgmt.	min costs	#items	#order	order/holding	entry-sales

Classification

Example	Objective	State	Action	Rewards	Dynamic
Inventory Mgmt.	min costs	#items	#order	order/holding	entry-sales
Airline Tickets					
Drinking at Party					
Exam Preparation					
Advertising					
Used Cars					
Eating Cake					

Classification

Example	Objective	State	Action	Rewards	Dynamic
Inventory Mgmt.	min costs	#items	#order	order/holding	entry-sales
Airline Tickets	max revenue	#tickets	#price	sales	current-sold
Drinking at Party	max fun	‰	#beer	fun/money	impact-rehab

Classification

Example	Objective	State	Action	Rewards	Dynamic
Inventory Mgmt.	min costs	#items	#order	order/holding	entry-sales
Airline Tickets	max revenue	#tickets	#price	sales	current-sold
Drinking at Party	max fun	‰	#beer	fun/money	impact-rehab
Exam Preparation	max mark/effort	#learned	#learn	effort, mark	learn-forget
Advertising	max profits	image	#advertise	campaigns	effect-forget

Classification

Example	Objective	State	Action	Rewards	Dynamic
Inventory Mgmt.	min costs	#items	#order	order/holding	entry-sales
Airline Tickets*	max revenue	#tickets	#price	sales	current-sold
Drinking at Party*	max fun	‰	#beer	fun/money	impact-rehab
Exam Preparation*	max mark/effort	#learned	#learn	effort, mark	learn-forget
Advertising	max profits	image	#advertise	campaigns	effect-forget
Used Cars	min costs	age	replace(y/n)	buy/repair	aging/faults
Eating Cake*	max utility	%cake	#eat	utility	outflow

* Finite horizon

General Problem Components

• What do you want to optimize (e.g., expected rewards)		(Objective)
• Define the state of your system		(State)
• Define the set of possible actions	(state dependent)	(Actions)
• Quantify event probabilities	(state+action dependent)	(Dynamics) (!)
• Define rewards	(state+action+event dependent)	(Rewards)
• Define state transitions	(state+action+event dependent)	(Transitions)
• What happens at the end (of the time horizon)?		(Final Rewards)

Recall - Questions?

- Finite/Infinite Time Horizon
- States
- Actions
- Events & Rewards
- Dynamics & State Transitions
- Deterministic/Stochastic
- Discrete/Continuous

Overview

Week	Dates	Topic	
1	**April** 12/**15**	Introduction + **Finite Time Markov Decision Processes (MDP)**	
2	April 19/22	Infinite Time MDPs + Dynamic Programming (DP) Exercise	
3	April 26/29	(Approximate) Dynamic Programming (ADP)	
4	May 3/6	Q-Learning (QL)	
5	May 10	Deep Q-Networks (DQN)	(Thu May 13 "Himmelfahrt")
6	May 17/20	DQN Extensions	
7	May 27	Policy Gradient Algorithms	(Mon May 24 "Pfingstmontag")
8	May 31/June 3	Project Assignments	
9	June 7/10	Work on Projects: Input/Support	
10	June 14/17	Work on Projects: Input/Support	
11	June 21/24	Work on Projects: Input/Support	
12	June 28/July 1	Work on Projects: Input/Support	
13	July 5/8	Work on Projects: Input/Support	
14	July 12/15	Final Presentations	
	Aug 31	Finish Documentation	